

팀 프로젝트 보고서

작성자: 임주희

1. 구현한 기술

- ① 풀페이지 적용
- ② 버튼 클릭 시 부드럽게 위로 스크롤
- ③ 스와이퍼 적용

2. 기술 설명

① 풀페이지 적용

- 우선 풀페이지가 적용되었으면 좋겠는 부분에 동일한 class명이나 id를 부여합니다.
저는 모든 페이지에 class='event'값을 부여했습니다.
- 그 뒤, 스크립트에 if문을 활용하여 '휠을 내리면' 과 '휠을 올리면'으로 나눈 뒤, 이중 if문을 활용하였습니다.

```
- $('.event').each(function () {  
-   $(this).on('wheel', function (evt) {  
-     evt.preventDefault();  
-  
-     let delta = evt.originalEvent.deltaY;  
-     let currentTop = null;  
-     const nextBox = this.nextElementSibling;  
-     const prevBox = this.previousElementSibling;  
-  
-     if (delta > 0) {  
-       if (nextBox) {  
-         currentTop = nextBox.offsetTop;  
-       } else {  
-         currentTop = this.offsetTop;  
-       }  
-     }  
-   })  
- })
```

```

-     } else {
-       if (prevBox) {
-         currentTop = prevBox.offsetTop;
-       } else {
-         return;
-       }
-     }
-   }
-
-   scrollTo({
-     top: currentTop,
-     behavior: 'smooth',
-   });
- });
- });

```

이렇게 이중 if문을 활용하면 휠을 내렸을 때, 다음 박스의 가장 위로 넘어가고, 다음 박스의 값이 없는 경우에는 멈추게 됩니다. 또한 휠을 올렸을 때는 이전 박스의 가장 위로 넘어가고, 다음 박스의 값이 없는 경우 멈추게 됩니다.

② 버튼 클릭 시 부드럽게 위로 스크롤

- 먼저 a태그의 기본 성격을 지워야 합니다.

```

- $('a[href="#"]').on('click', (evt) => {
-   evt.preventDefault();
- });

```

- 그 다음, 'bottom-btn'을 클릭하면 top:0으로 'smooth'하게 올라가도록 스크립트를 작성해줍니다. 그럼 'bottom-btn'을 클릭하면 가장 위로 부드럽게 올라가게 됩니다.

```

- $('.bottom-btn').on('click', () => {
-   scrollTo({
-     top: 0,
-     behavior: 'smooth',
-   });
- });

```

③ 스와이퍼 적용

- 스와이퍼의 경우 '<https://swiperjs.com/>'에 접속하면 사용 방법이 자세하게 나와있습니다. 우선, swiper의 기본 css와 js를 다운받아 html에 연결시켜주어야 합니다.

```
- <!-- swiper css -->
- <link rel="stylesheet" href="../common/swiper-bundle.min.css">
- <!-- swiper -->
- <script src="../common/swiper-bundle.min.js"></script>
```

- 그 다음 html의 경우 swiper에서 제공하는 기본적인 html구조를 따르며 작성을 해줍니다. 이때, 원만한 css 수정을 위해 class명이나 id 값을 본인이 원하는 값도 추가로 넣어줍니다.

```
- <div class="event-banner event">
-   <div class="swiper mySwiper1">
-     <div class="swiper-wrapper">
-       <div class="swiper-slide bg1">1
-         <div class="view-more">
-           view more
-         </div>
-       </div>
-
-       <div class="swiper-slide bg2">2
-         <div class="view-more">
-           view more
-         </div>
-       </div>
-
-       <div class="swiper-slide bg3">3
-         <div class="view-more">
-           view more
-         </div>
-       </div>
-     </div>
-     <div class="swiper-button-next"></div>
-     <div class="swiper-button-prev"></div>
-     <div class="swiper-pagination"></div>
-   </div>
```

```
- </div>  
-
```

- Css의 경우, 원하는 이미지와 값들을 입력해주면 됩니다. 저는 전체적인 사이즈와 background-image를 입력해주었습니다.
- Js의 경우 swiper에서 제공하는 기본 js를 가져와 본인의 디자인에 맞게 수정하면 됩니다.

```
const swiper = new Swiper('.swiper', {  
  // Optional parameters  
  direction: 'vertical',  
  loop: true,  
  
  // If we need pagination  
  pagination: {  
    el: '.swiper-pagination',  
  },  
  
  // Navigation arrows  
  navigation: {  
    nextEl: '.swiper-button-next',  
    prevEl: '.swiper-button-prev',  
  },  
  
  // And if we need scrollbar  
  scrollbar: {  
    el: '.swiper-scrollbar',  
  },  
});
```

저는 이 기본 틀에서 자동으로 넘어가는 것을 구현하기 위하여 autoplay나 direction 값을 수정하였습니다. 추가로 원하는 값들은 직접 수정해주면 됩니다.

- 추가로 기본 css의 값이 적용된 것을 바꾸고 싶은 경우 개발자 도구에 들어가 해당 값을 확인하여 직접 수정하면 됩니다.

3.문제점

- ① 풀페이지 적용
 - 크게 html의 구성이 nav-contents-footer방식이었는데, contents영역은 풀페이지 적용이 잘 되지만, footer 페이지는 풀페이지가 적용되지 않고 스크롤도 넘어가지 않는

문제점이 생겼습니다.

② 부드럽게 위로 스크롤

- 간단하고, 자주 사용했던 기술이기에 문제점은 없었습니다.

③ 스와이퍼 적용

- 스와이퍼 사이트에서 기본 틀을 제공하기에 문제점은 없었습니다.

4.문제 해결

④ 풀페이지 적용

- Html의 구성을 nav-contents-footer방식이 아닌, contents 안 footer를 넣으며 수정을 하니, 모든 페이지에 풀페이지가 정상적으로 작동했습니다.

⑤ 버튼 클릭 시 부드럽게 위로 스크롤

- 간단하고, 자주 사용했던 기술이기에 문제점은 없었습니다.

⑥ 스와이퍼 적용

- 스와이퍼 사이트에서 기본 틀을 제공하기에 문제점은 없었습니다.